



# Self-Triggering at ProtoDUNE

Thijs Miedema

Supervised by Frank Filthaut

# Introduction



- Last week some Phil presented the next step at ProtoDUNE towards self-triggering
- We look forwards to the step after with some notes on infrastructure

# Goals



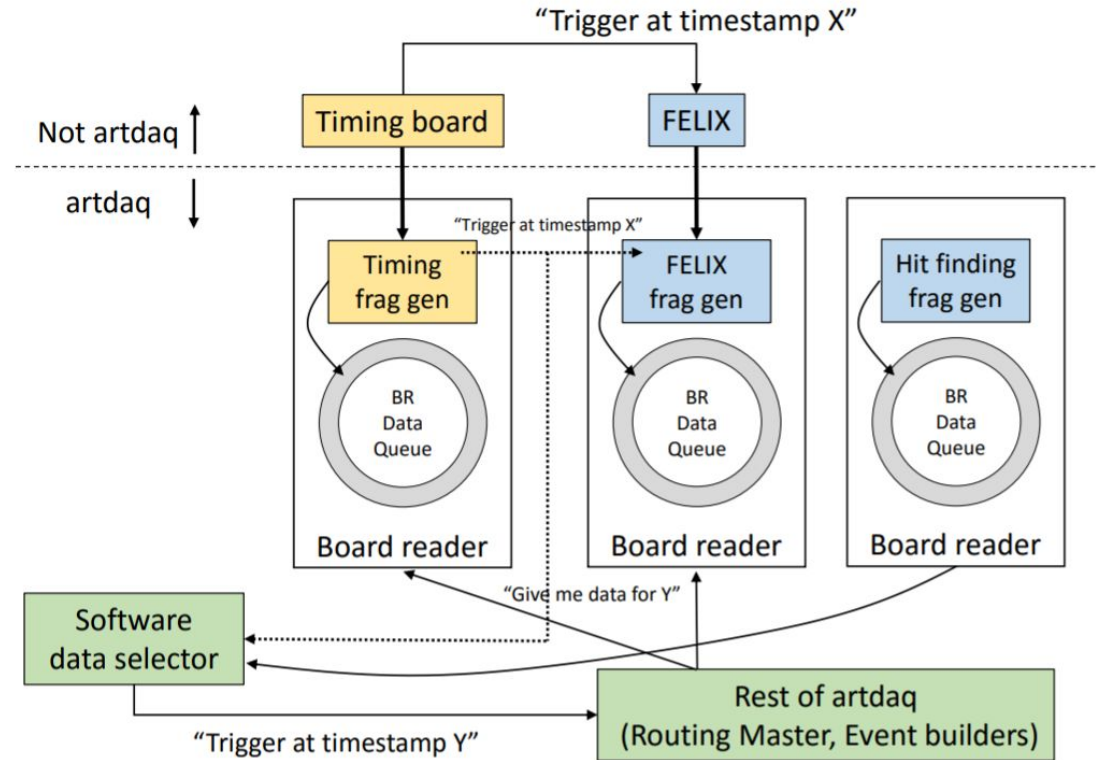
- A self-triggering ProtoDUNE in 2019
- Requires actual trigger, not just primitives/candidates, need to define interesting/calibration signals to not get swamped with all cosmics
  - Horizontal cosmics
  - Stopping cosmics

# Concrete plan



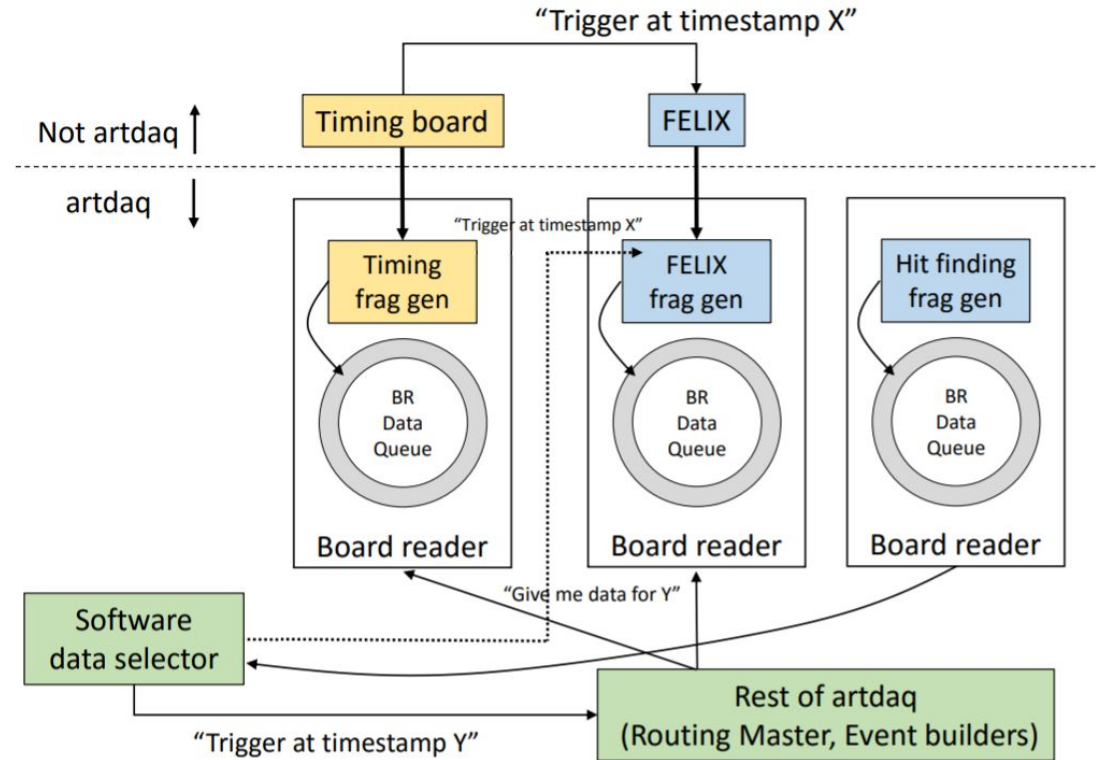
- Work together in producing trigger candidates in C++ realtime at ProtoDUNE
- Take these candidates and use them to create algorithms to select our interesting signals, again, realtime at ProtoDUNE
- Heavily rely on Brett's IPC to ship our data:
  - Nice to evaluate working with this
  - Allows us to move our “algorithm boxes” from physical box to other physical box

# Phil's self-triggering design



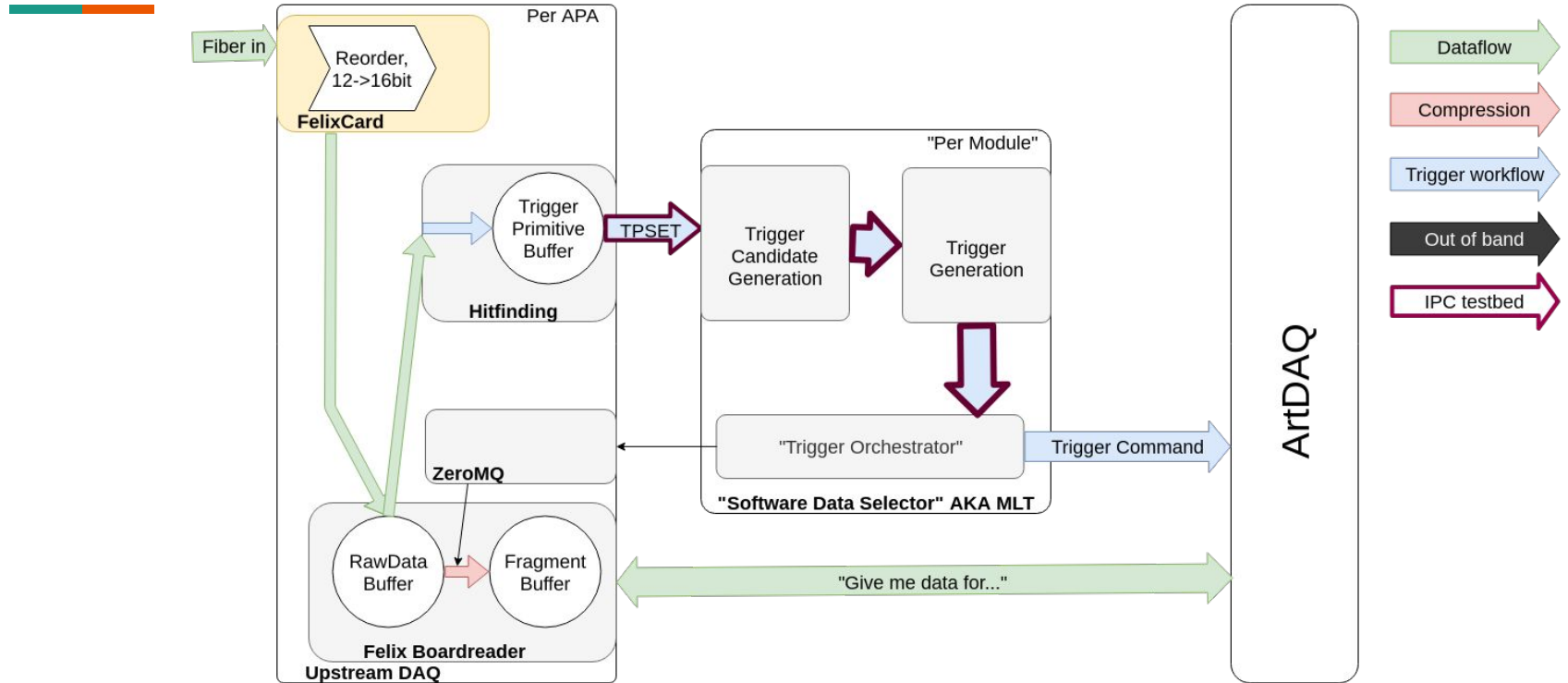
- Minimal changes with respect to current setup to allow a form of self-triggering
- Main drawback: can only trigger within existing timewindows as spit out by the timing board+configuration
- This is definitely not how it will work @DUNE

# Phil's self-triggering design++

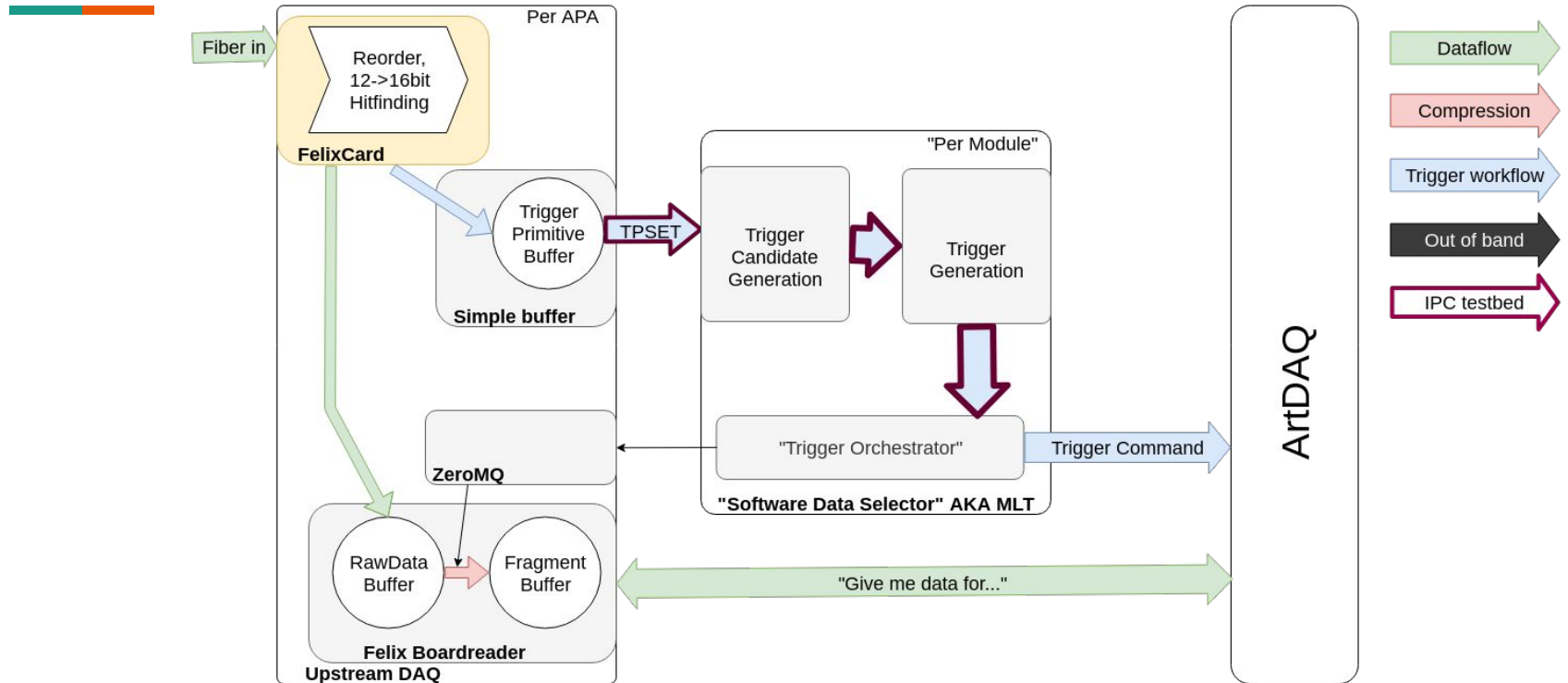


- Instead of the timing board giving the ZeroMQ command to build the fragment we let the software data selector do it.
- Send the trigger to artdaq with slight (few ms) delay to allow the frag gen to do its thing
- Still not how we want it at DUNE, but closer
- Allows full self-triggering

# Yet Another Attempt At Putting This In A Diagram

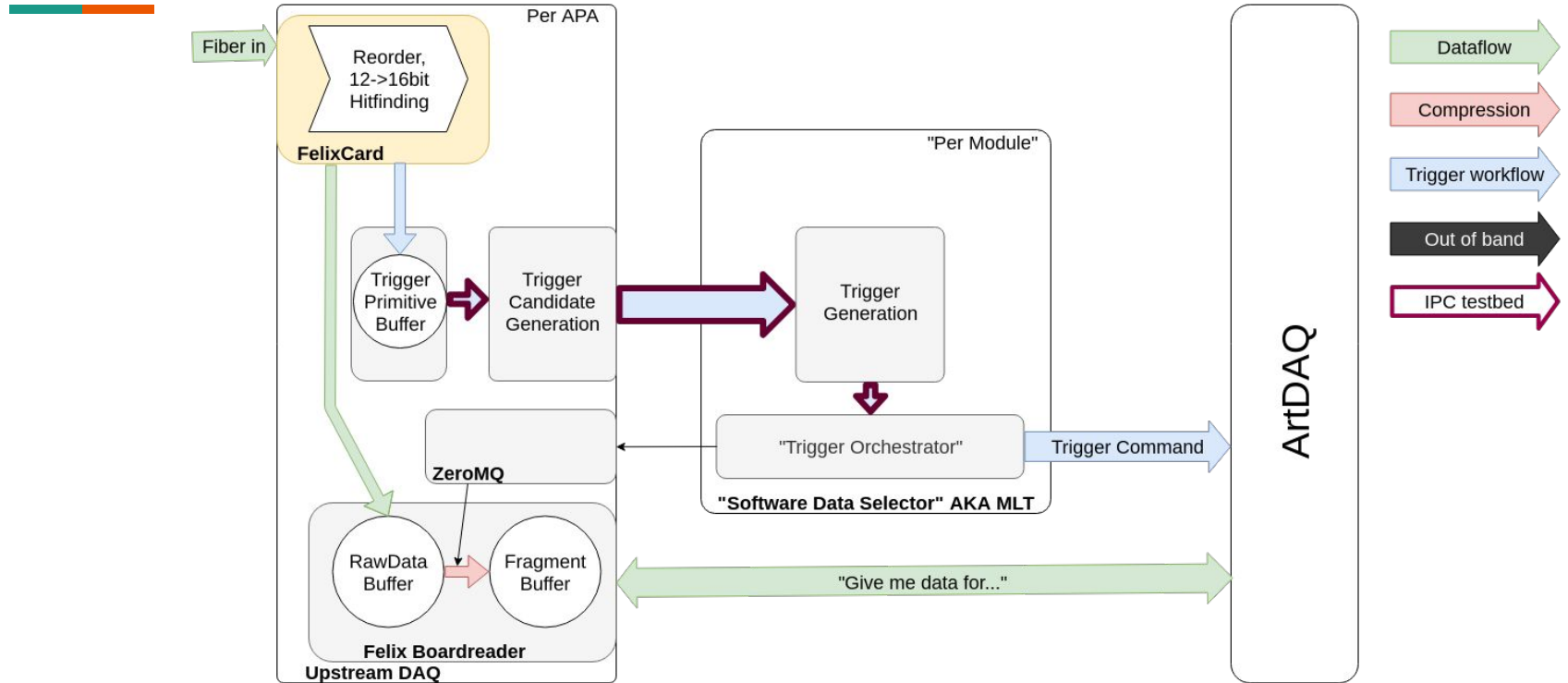


# Y.A.A.A.P.T.A.





# Y.A.A.A.P.T.A.



# Concluding remark



- With minimal changes to existing systems we can build a DUNE-like triggering system at ProtoDUNE